

2015

# Evaluation of Two-Dimensional Codes for Digital Information Security in Physical Documents

Shuai Chen

*University of Massachusetts Amherst*

Follow this and additional works at: [https://scholarworks.umass.edu/masters\\_theses\\_2](https://scholarworks.umass.edu/masters_theses_2)



Part of the [Digital Communications and Networking Commons](#)

---

## Recommended Citation

Chen, Shuai, "Evaluation of Two-Dimensional Codes for Digital Information Security in Physical Documents" (2015). *Masters Theses*. 186.

[https://scholarworks.umass.edu/masters\\_theses\\_2/186](https://scholarworks.umass.edu/masters_theses_2/186)

This Open Access Thesis is brought to you for free and open access by the Dissertations and Theses at ScholarWorks@UMass Amherst. It has been accepted for inclusion in Masters Theses by an authorized administrator of ScholarWorks@UMass Amherst. For more information, please contact [scholarworks@library.umass.edu](mailto:scholarworks@library.umass.edu).

**EVALUATION OF TWO-DIMENSIONAL CODES FOR  
DIGITAL INFORMATION SECURITY IN PHYSICAL  
DOCUMENTS**

A Thesis Presented

by

SHUAI CHEN

Submitted to the Graduate School of the  
University of Massachusetts Amherst in partial fulfillment  
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL AND COMPUTER ENGINEERING

May 2015

Electrical and Computer Engineering

**EVALUATION OF TWO-DIMENSIONAL CODES FOR  
DIGITAL INFORMATION SECURITY IN PHYSICAL  
DOCUMENTS**

A Thesis Presented

by

SHUAI CHEN

Approved as to style and content by:

---

Tilman Wolf, Chair

---

Michael Zink, Member

---

David Irwin, Member

---

C.V.Hollot, Department Head  
Electrical and Computer Engineering

## ACKNOWLEDGMENTS

First of all, I will give thanks to Professor Tilman Wolf for his guidance during the project and his instructions on my thesis work. Also thank my committee members Professor Michael Zink and Professor David Irwin, for their kind help in my thesis proposal and defense.

Secondly, I would like to thank my team members Pengcheng Wang, Xue Yu, Shuo Guo and Padmaja Duggisetty for their contributions in the project. Especially thank our team leader Pengcheng, he groups us together and makes us work under team spirit.

Also, I want to thank my lab mates, PhD student Xinming Chen, PhD student Thiago Teixeira, and PhD student Hao Cai for their advice on my project.

Last but not least, I want to thank my girlfriend Yuan Liu, and my parents for their encouragement and love.

## ABSTRACT

# EVALUATION OF TWO-DIMENSIONAL CODES FOR DIGITAL INFORMATION SECURITY IN PHYSICAL DOCUMENTS

MAY 2015

SHUAI CHEN

B.Sc., CHONGQING UNIVERSITY

M.S.E.C.E., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Tilman Wolf

Nowadays, paper documents are still frequently used and exchanged in our daily life. To safely manage confidential paper information such as medical and financial records has increasingly become a challenge. If a patient's medical diagnosis get stolen or dumped without shredding, his or her private information would be leaked. Some companies and organizations do not pay enough attention to the problem, letting their customers suffer the loss. In the thesis, I designed a hybrid system to solve this problem effectively and economically. This hybrid system integrates physical document properties with digital security technology, which brings in a revolutionary idea for processing sensitive paper information in modern world. Based on that, I focus on different QR code sizes and versions, compare their attributes and relations, and find the best QR code size and version according to data amount in a given area. Finally I implement them in CryptoPaper word plugin, using several test cases to test the functionality of it.

# TABLE OF CONTENTS

	Page
<b>ACKNOWLEDGMENTS</b> .....	<b>iii</b>
<b>ABSTRACT</b> .....	<b>iv</b>
<b>LIST OF TABLES</b> .....	<b>vii</b>
<b>LIST OF FIGURES</b> .....	<b>viii</b>
 <b>CHAPTER</b>	
<b>1. INTRODUCTION</b> .....	<b>1</b>
1.1 Background .....	1
1.2 Motivation .....	2
1.3 Problem statement .....	4
1.4 Contribution .....	5
1.5 Organization .....	5
<b>2. BACKGROUND AND RELATED WORK</b> .....	<b>6</b>
2.1 System Architecture .....	6
2.2 System functionality .....	8
2.3 System design and Implementation .....	9
2.3.1 Two Dimensional Code .....	9
2.3.2 CryptoPaper Codes .....	11
2.3.2.1 Why choose QR-code .....	11
2.3.2.2 What is QR-code .....	11
2.3.3 Meta-information .....	13
2.3.4 Key management database .....	14
2.3.5 Data recovery .....	16

<b>3. PROJECT PLAN</b> .....	<b>18</b>
3.1 CryptoPaper generation working flow .....	18
3.2 QR code choices .....	20
3.2.1 QR-code properties .....	20
3.2.1.1 High data capacity .....	20
3.2.1.2 High data density .....	21
3.2.1.3 Data restoration .....	21
3.2.1.4 Omni-directional and high-speed reading .....	22
3.2.1.5 Structure Appended Feature .....	22
3.2.1.6 Multiple version and size choices .....	23
3.2.2 QR-code experiments .....	25
3.2.2.1 Printer Head Density and Module size .....	25
3.2.2.2 Scanner factor .....	25
3.2.2.3 QR code decision factors .....	26
3.2.3 Experiments result analysis .....	27
3.2.4 Implementation in MS Word add-in .....	32
3.2.4.1 Word Object Module in a developer's perspective .....	32
3.2.4.2 Introduction to the QR-code library in c# .....	34
3.2.4.3 Select the best choices among trade-offs .....	35
3.3 AES encryption .....	36
3.4 Microsoft MySQL database connectivity .....	37
3.5 Password hash .....	38
<b>4. SYSTEM TEST AND EVALUATION</b> .....	<b>41</b>
<b>5. SUMMARY</b> .....	<b>46</b>
<b>BIBLIOGRAPHY</b> .....	<b>47</b>

## LIST OF TABLES

<b>Table</b>	<b>Page</b>
3.1 QR code data storage capacity .....	21
3.2 QR code data density .....	21
3.3 QR code error correction level .....	22
3.4 Printer dpi and module size .....	26
3.5 Test distance for 0.5mm module size with level L .....	28
3.6 Test distance for 1.0mm module size with level L .....	28
3.7 Test distance for 0.5mm module size with level H .....	29
3.8 Test distance for 1.0mm module size with level H .....	29
3.9 QR code version for fixed data amount of different ECC levels .....	35



## LIST OF FIGURES

Figure	Page
1.1 Paper encryption technology system operation . . . . .	2
2.1 CryptoPaper system architecture. . . . .	6
2.2 CryptoPaper system operation. . . . .	8
2.3 Code 49 . . . . .	10
2.4 PDF 417 . . . . .	10
2.5 Data matrix barcode . . . . .	10
2.6 QR code . . . . .	10
2.7 QR-code composition . . . . .	12
2.8 Key management database structure. . . . .	15
2.9 Recovering information with PC-based scanner. . . . .	17
3.1 Word plugin operation process. . . . .	19
3.2 Position detection patterns to enable the omni-directional scan . . . . .	22
3.3 Structure Append Feature . . . . .	23
3.4 QR-code version properties. . . . .	23
3.5 Data bits in QR codes of different versions. . . . .	24
3.6 Number of QR codes that fit in space of version-40 QR code. . . . .	24
3.7 Data bits in area of version-40 QR code (theoretical and practical when read with scanner). . . . .	24

3.8	QR code generation factors .....	27
3.9	Relation between version and size for different modules .....	30
3.10	Relation between version and scan distance for different modules .....	30
3.11	Relation between scan distance and size for module 1 .....	31
3.12	Relation between scan distance and size for module 2 .....	31
3.13	Relation between scan distance and size for module 1 .....	32
3.14	Relation between scan distance and size for module 2 .....	32
3.15	The Application object contains the Document, Selection, Bookmark, and Range objects. ....	33
3.16	AES encryption in ECB mode. ....	36
4.1	Test case for different font size .....	41
4.2	Font size test result .....	41
4.3	Test case for long paragraphs .....	42
4.4	Test case for long paragraphs .....	42
4.5	Test case for data overflow .....	43
4.6	Generated employee information form using word add-in .....	44
4.7	Test result for access control of CryptoPaper system .....	44

# CHAPTER 1

## INTRODUCTION

### 1.1 Background

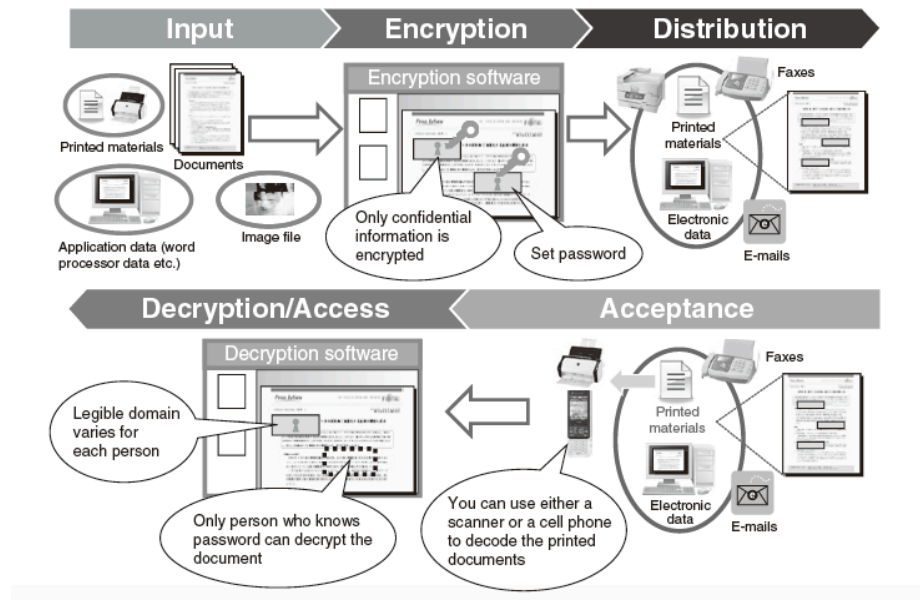
Even in the information age, paper documents are still widely used in multiple areas such as government, education, and business. This traditional way of carrying information has several advantages that cannot be replaced by e-files. Paper is convenient to use and modify. It cannot be disseminated everywhere, which is unlike digital documents, once open to the public, authors could hardly control the dissemination. In addition it has high accessibility; people can put it in the bag or pocket and take it everywhere without worrying about losing power or file corruption.

However, paper documents, if not properly managed, could lead to information leakage as easily as electronic files, which would result in serious consequences such as identity theft and privacy invasion. In 2013 January, a medical group was fined \$140,000 for tossing patients' health records into public dumpster without shredding or redacting. In 2010, a company suffered a \$50,000 penalty for tossing consumers' credit report information into unsecured dumpster. Security industry has constantly been trying to address the need to protect confidential information by building stronger information privacy protocols, especially in the field of protecting digital information. While people pay great attention to the security of their digital documents, paper documents security is overlooked. Currently physical documents containing private information can either be stored safely in the cabinet or be shredded before dumping.

## 1.2 Motivation

Usually companies and organizations either shred paper documents before dumping or store them in the cabinet. Shredding service is simple, secure and environmentally friendly. But for some smaller companies without a large amount of materials to be shredded, hiring a company to do the job is an extra expense the business may not be able to afford. Storing paper documents in the cabinet is another alternative. It is convenient and cheap, however it is space consuming and had to manage. Also, documents are under risks of being stolen or destroyed if they are not stored properly.

The idea of paper encryption technology has been pioneered by several Japanese scientists from Fujitsu Laboratories. Hereby I briefly describe the ideas of them in their paper[1].



**Figure 1.1.** Paper encryption technology system operation

As shown in Figure 1.1, they transfer source data (in digital format or physical format) into image file first. Then the user could select the information area in the image file, and enter a password to encrypt it. The image data that has been encrypted may be converted and saved as electronic data again or printed. It can be

sent to the recipient by post, fax, or by email. Once the recipient open the document with scanner, he or she can input the password to recover the data.

They applied several user scenarios for paper encryption technology. First, some documents such as fault control sheets, corporate governance and internal audit data, and court disclosure need to be prepared anew before being submitted in order to keep confidential information away from the public. By using this technology, original copy can be partially encrypted and the time spent on preparing a new version can be saved. Second, it can reduce labor of hiding confidential information in paper documents with a black mark. Third, encryption technology makes it possible to disclose minimal amount of information necessary for a company using outsource services. In this way, companies protect their confidential information from leaking into contractor companies.

However, this application has its limitations as well. All the paper documents need to be transferred into image data before and after encryption. The data can be easily damaged by 3D distortions, optical noise and blurring. These issues are hard to address simply by developing image processing algorithms. Although, a higher resolution of camera ensures a better image quality can be obtained after decryption, however, it takes longer processing time.

Our system, called CryptoPaper, borrows the idea from that paper, but overcomes its limitations by encrypting and transferring portions of a physical document into machine-readable code before printing. Besides, a cloud database with a role-based system will take care of key management, so that only the authorized user can get access to the key and recover original data. Based on that, this paper studies the generation and detection technology of two-dimensional code[2], implements different attributes of QR code in our system, and evaluates their performances.

### 1.3 Problem statement

There are a list of technical challenges to design and implement this CryptoPaper system. This section states these challenges and proposes assumptive solutions respectively, and detail solutions will be discussed in the next chapters.

The first issue is how to reduce the cost of implementation. Now exciting services such as tamper-evident paper, physical access control or proper shred are expensive to implement. In the CryptoPaper system, for the encryption part, users can use a word plugin with multiple functions for free. However, for the decryption part, they have to use a computer-based scanner to recover the original data, which costs money to buy. A better planning would be to build an mobile app on smartphones in the given time. This allows users to use update-to-date services without owning a scanner by themselves. In this way, people only need to pay service fees instead of spending money on hardware devices.

The second issue is limited ability to revoke access. Once documents are issued, it is difficult to revoke its access. Attackers can take advantage of this soft spot and get the information from inside and gain profit from selling it. We try to address this challenge by prohibiting the improper handling of paper documents and allowing only authorized users to have access to the data in an economic way. In other words, we use a hybrid way to realize confidentiality, integrity and authenticity on paper documents. Finally, how to protect privacies of people once database gets hacked is another challenge. Our assumption is only store the encryption key on the cloud database, and keep the original data on the paper. In this way, hackers will not get real information even they hack the database. People can still protect their data by destroying paper documents or revoking the access key. There are some more trivial problems to be concerned, I will discuss them in next few chapters as well.

## 1.4 Contribution

CryptoPaper system encrypt and transfer portions of a physical document into machine-readable code before printing. A scan device with suitable image recognition technology is used to read the code and recover the encoded data. A cloud-based database is used to implement role-based access control, in order to ensure only authorized users can interpret the data. This paper fills the gap by focusing on realizing a practical system that combines digital security technology with traditional paper. It inherits most of the advantages of traditional paper while overcomes the disadvantages of paper shredding service and storage. Paper lost would no longer be a concern because of access control. It is affordable and user friendly, thus can be appropriately applied to most of the situations for company use or personal use. Then I will implement different 2D codes in the system, compare their attributes such as data capacity, data density, and detecting speed to testify why QR code is the best choice for our system.

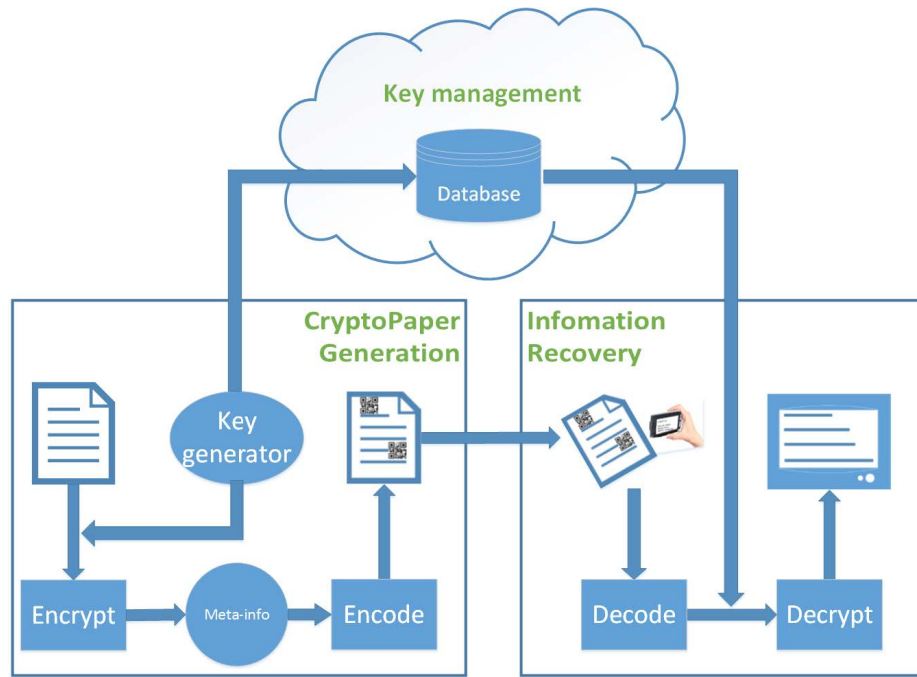
## 1.5 Organization

The remainder of the thesis is organized as follows: Chapter 2 presents a brief introduction to CryptoPaper system including system architecture and functionalities. The definition of two-dimensional codes and different kinds of 2D codes is also reviewed in this chapter. Then in Chapter 3 I present detailed project plans used to realize the system. In Chapter 4, I present the evaluation and performance result of the system. Finally, this paper concludes in Chapter 5 with system summaries and suggestions for future research.

## CHAPTER 2

### BACKGROUND AND RELATED WORK

#### 2.1 System Architecture



**Figure 2.1.** CryptoPaper system architecture.

System architecture of CryptoPaper is illustrated in Figure 2.1. There are three modules in the creation and use of CryptoPaper system:

1. Creation of document: We use a software (in our case a Microsoft Plugin) to encrypt sensitive information and replace cipher text with 2-dimensional code (in our case, QR code). User needs to select the plaintext range first by clicking on the transformation button in our plugin; then the software will

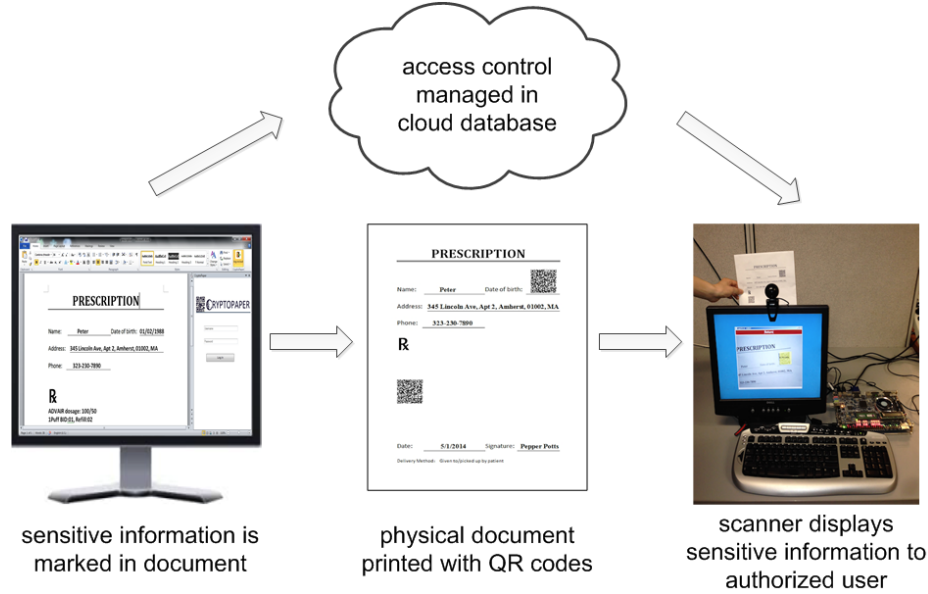


encrypt plaintext together with meta-information using a key generated by the random key generator, and transfer it into QR-code. After the user set up access control, software will upload the key into cloud database. User should not be concerned about their private information getting intercepted, with normal QR-code reader. What's more, both CryptoPaper generation software and data recovery scanner are made available only to the registered users. A secure login system is used to take care of this on two terminals.

2. Access control management: A cloud-based database serves as a backbone to both generation and recovery modules. No any actual data is uploaded into database. It is used to store key and user information. Scanner can retrieve the key from database according to the code id in the meta-information of every QR-code. User information includes the organization individual belongs to, role of individual in the organization, user's access rights, use's name and address, etc. When someone encodes paper document, he or she can choose determine who the authorized users would be so that accessibility of information is strictly and objectively controlled. Once the end user log in with username and password on scanner machine, the corresponding access right can decide whether he or she can recover the information on the paper documents.
3. Sensitive information recovery: Encoded information on a physical document can be recovered by software on scanner. The user is required to login into the CrptoPaper system and scan the codes in document. If the end user's access right matches with the predetermined access right of code id, the user can view the sensitive data instantly on scanner screen, else he or she will see "Sorry, you are not authorized" instead.

## 2.2 System functionality

Figure 2.2 shows the CryptoPaper system operations. Our system provides the following functionalities with different features:



**Figure 2.2.** CryptoPaper system operation.

- User modifies paper document in Microsoft Word, replace the sensitive information with QR code, set up access right for the end user and upload the key. After that, he or she can print paper document out, and hand it to other people. Sensitive information is encrypted by AES algorithm and only stored in the code on physical document. In this way, even attackers breach the cloud database and get the key of each code they still cannot get the privacy data unless they have physical document as well.
- Once the use get the paper document, he or she can log-in CryptoPaper system using our scanner to recover the data. Only authorized users can access encoded information since both the physical document and access right to the digital key are required. Authors can set up the access right of CryptoPaper before printing and revoke the access right of paper anytime.

- In the future, we can use asymmetric encryption algorithm with digital signature, create public private key pairs for each user of the system and ensure the integrity and authenticity of the system.

## 2.3 System design and Implementation

### 2.3.1 Two Dimensional Code

Two-dimensional (2D) barcodes provide a means of embedding Web addresses, text or other data in a camera-readable format. 2D barcodes enables users to scan a code with their form and automatically directed to a web page or information contained within the code without having to remember. As mobile devices outpace desktops as the primary means of accessing Internet, 2D barcodes come into a wide range popularity.

In two-dimensional barcodes many thousand alphanumeric characters can be placed in single symbol[3]. In the two-dimensional symbols, data are encoded in both the height and width of the symbol, and the amount of data that can be contained in a single symbol is significantly greater than that stored in a one dimensional symbol. And one of the amazing aspects of two-dimensional symbols is their potential durability to make them can be read easily and write accurately. Overall, a two-dimensional barcode is a graphical image hat stores information both horizontally and vertically. It contains more data amount and better durability than 1D code.

In general the 2D barcode are classified in two groups which are stacked 2D barcodes such as code 49 and PDF417, and matrix 2D barcodes, such as data matrix barcode and QR code.

Stacked 2D barcodes: Code 49 is a primitive type 2D code. It is developed to fill a need to pack a lot of data in very small symbol. Code 49 accomplishes this by using series of barcode symbols stacked one on top of another. It can encode complete ASCII 128 code. Figure 2.3 is an example for Code 49.

PDF 417 capabilities include symbol can link to others symbols which are scanned in sequence allowing even more data to be scored. Data codeword zone are group or bars and spaces representing one or more numbers, letters and other symbols. Figure 2.4 is an example for PDF 417.



**Figure 2.3.** Code 49



**Figure 2.4.** PDF 417

Matrix 2D barcodes: The data matrix barcode is high-density, two dimensional symbology that encodes the number, files, and actual data bytes. The characters, numbers, text, and actual bytes of data may be encoded including Unicode characters and photos. The valid characters for data matrix barcode are ASCII 0 to 255. Figure 2.5 is an example for data matrix barcode.

QR code has higher data density and capacity support kanji/Chinese character. Figure 2.6 is an example for QR code. In this project, I will focus on QR codes and generate the best choice code for users in our word add-in based on different QR code attributes. Details of QR code will be discussed in the next section.



**Figure 2.5.** Data matrix barcode



**Figure 2.6.** QR code

## 2.3.2 CryptoPaper Codes

### 2.3.2.1 Why choose QR-code

The two-dimensional code used to store important information in CryptoPaper system has several property requirements. The code must be machine-readable with sort of popularity in real life. It should have sufficient density to store enough data. Last but not least, it can be generated in an easy manner. The following list some advantages that QR-code has over other two-dimensional codes. Overall, QR codes have all good features of other 2D symbols:

- High data capacity
- High speed reading
- High density recording

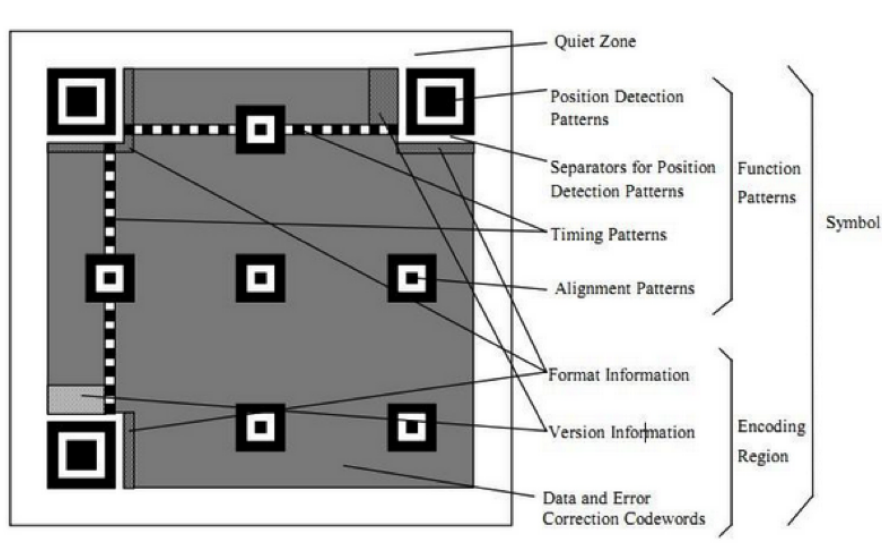
Individually, it has lots of special qualities like the ones listed below:

- High error correction (7%-30%)
- High data density
- Small printout size
- Various versions (1-40)
- Readable from any direction in 360 degrees

Considering all these features, we choose QR codes for encoding data with meta-information.

### 2.3.2.2 What is QR-code

QR stands for “Quick Response”. QR-code is a 2 dimensional bar code that was created by the Japanese corporation named Denso-Wave in 1994 and is aimed at decoding contents at high speed. Since the QR code is widely used today and is a



**Figure 2.7.** QR-code composition

standard and mature technology to encode and decode information, using QR code to carry the encrypted information will be more convenient and reliable than using other form of code. What is more, a QR code is easier to read with a device. Thus, this method can be used for secure document delivery in both electronic and paper form, and the encryption document can be recovered quickly and accurately. Figure 2.7 lists the composition of QR code. QR code basically consists of black cells arranged in a sequence grid on a white background in different zones. Quiet zone leaves a quiet space for QR code; function pattern used to locate the position of QR code; encoding region stores the column information and some other additional data; data and error correction codewords stores the real data inside.

There are several configuration options for QR codes. The information encoded by a QR code can be made up of four standard types ("modes") of data (numeric, alphanumeric, byte/binary, Kanji), and through supported extensions almost any type of data can be encoded. The QR-code system along with fast readability, can store a large amount of data in a single code. Following lists parameters to set up during the creation of QR code.

Encoding modes: QR codes can be encoded in numeric (0-9) alphanumeric (0-9 A-Z etc.), 8-bit bytes, and Kanji characters. In our project, we chose alphanumeric mode.

Error correction: It is defined in four levels, namely, 7 percent or less (Level L), 15 percent (Level M) or less, 25 percent (Level Q) or less and 30 percent (Level H) or less.

Version: it varies from 1 to 40. Version 1 is written as 21 X 21 and can store 128bits and when the version increases, the size of code modules increase by 4 in each dimension, along with the amount of data that can be stored. The version 40 is denoted as 177 X 177 in 18672 bits.

The amount of data that can be stored in the QR code symbol depends on the data type (mode, or input character set), version (1, 2,..., 40, indicating the overall dimensions of the symbol), and error correction level. The maximum storage capacities occur for 40-L symbols (version 40, error correction level L), which can hold as many as 23648 data bits. If the contexts on it are in general format (Calibri, font size 11, 1.5-line spacing), there should be about 19400 bits per page. Thus the data density for an A4 paper document in general format is about 200 bits per inch<sup>2</sup>. For comparison, for version-1 QR code (72 bits, 0.4225 inch<sup>2</sup>) printed with 1200dpi printer and 37-dot configuration, the data density will be 170 bits per inch<sup>2</sup>. For version-2 (128 bits, 0.5625 inch<sup>2</sup>), the data density is 228 bits per inch<sup>2</sup>. And when the version of QR code is more than 2, the data density of QR code is more than that of paper document. Therefore, all the data on paper document can be put into QR codes with less area, which implies a high data density.

### **2.3.3 Meta-information**

In simple terms, metadata is “data about data”, and if managed properly, it is generated whenever data is created, acquired, added to, deleted from, or updated in

any data store and data system in scope of the enterprise data architecture. Metadata can provide clarity of relationships and clarity of data lineage, and remove inconsistency and redundancy. In our CryptoPaper document, we create four elements for our metadata.

- Code ID records the number for each QR-code be created and saved as an index in our database to enable decryption of the appropriate decryption of code.
- The second one is the cipher text, which contains the sensitive data and can only be viewed by the authorized user.
- The third one is digital signature, hashing the cipher text with producer's private key. It can resolve the inconsistencies of paper document in case anyone tampers the encrypted information on the paper.
- The fourth one is the public key, it is necessary for end user to verify the digital signature of the paper.

#### **2.3.4 Key management database**

We use Amazon Relational Database Service (Amazon RDS) to build our database, which is a web service that makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, freeing developers up to focus on applications and business.

Our system use role-based access control to decide weather a user can recover a code or not. Each registered organization has many different roles for different people, and each people have their role in the organizations that they belong to. This section introduces the relational database work behind. Figure 2.8 provides an overview of the database schema.



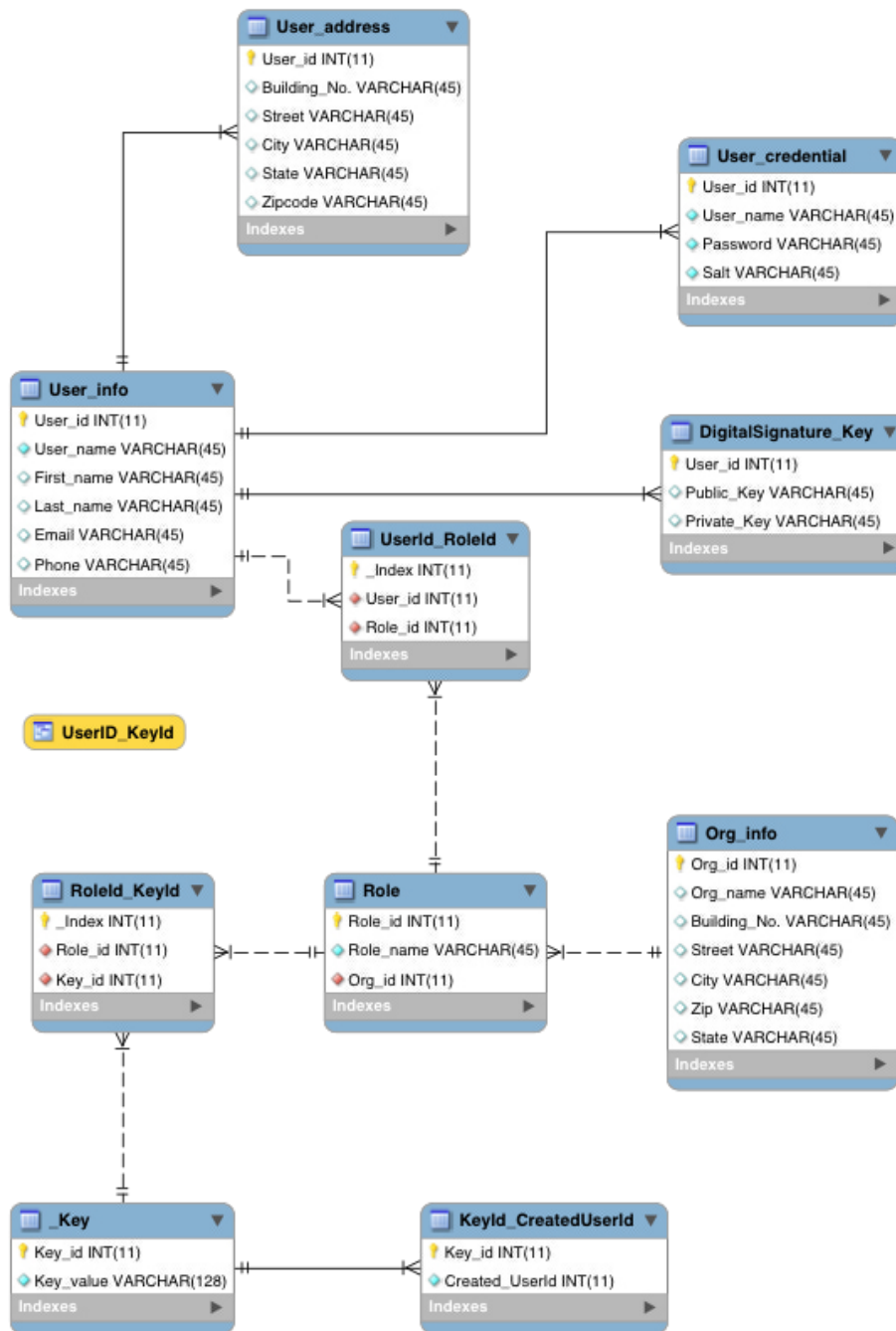


Figure 2.8. Key management database structure.

There are ten tables in the schema. User\_info table is used to store personal information. User\_id works as primary key in this table and foreign key in four other tables. Every user has one user\_id, it is exact the same with their username in the system. Role\_id is combined with key\_id in one table for query. When end user login with scanner, our system can get his role\_id according to his/her user\_id. If user's role\_id matches with the role\_id with each code, our system will query the key value from another table. To ensure the authenticity and integrity, digital signature function will be introduced in the future. In this way, a table will be used to store the user\_id of who create this document with different key\_id.

Besides, if the user selected wrong access right for specific code id, he/she can revoke it even after printing documents at any time. And if some employee leaves the organization he/she worked before or changes the role in that organization, database maintainers can change the role of people in the database for synchronization.

### **2.3.5 Data recovery**

For information recovery part, my teammate creates a login system using Java for user authentication. This Java application has a graphical user interface which provides verification for user identities and allows them to log into the system. Once they login to our system successfully, the software then opens the webcam and displays another frame to display the content being scanned by the webcam. Next the webcam will check whether there are any QR codes in the content scanned from the document. If there are, it will obtain the meta-information from id and try to decode it. User's identity will be checked with access right of the document. If he/she is authenticated, it will retrieve key from the database and then use it to decrypt the cipher text. After getting the plaintext, it will cover the QR codes in the video with semi-transparent rectangles and display plaintext on these areas. Otherwise, captions like "Sorry, you are not authorized" will show on the screen. Figure 2.9 shows the result of recovering

CryptoPaper information. Based on that, we also designed a scanner application on smart phones, it supports a better user experience with the same functionalities.



**Figure 2.9.** Recovering information with PC-based scanner.

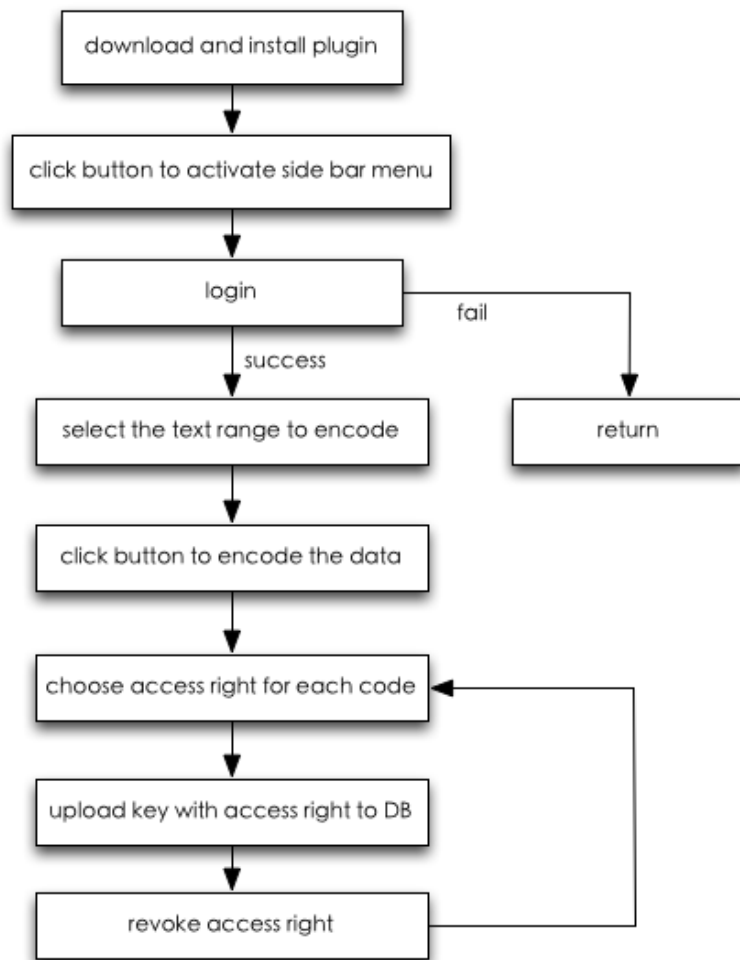
## CHAPTER 3

### PROJECT PLAN

#### 3.1 CryptoPaper generation working flow

In this section I describe the steps in creating a CryptoPaper i.e., creating an encrypted document from a plain text document. The steps are described as Figure 3.1:

1. Initially download and install the CryptoPaper plugin to the Microsoft word, open the required document to be encrypted or create a new document and enter the text to be encrypted.
2. Click on “CryptoPaper” toggle button on Microsoft Word toolbar to activate a sidebar.
3. Login to access the CryptoPaper software. You will be prompted to provide your user-id and password after registered. We currently don't provide any self-register function for new users. If a new user want to register in, he or she could provide the details required by our system including personal information along with role, and organization that belongs to. Our system will keep that in the clod database management system and create an account for that user. Once the username and password match the details present in the CryptoPaper database, the user will be granted access to the system else he or she will be denied.



**Figure 3.1.** Word plugin operation process.

4. If the user is allowed to login, the sidebar will direct the user into next menu, which contains the list of organizations and roles with several buttons. User can choose the organization and roles as to who can view which piece of information.
5. The User can then select the sensitive data that he/she wants to encrypt and click on the “Encode” button. A beautiful QR-code with the corresponding version and data amount will be displayed on the screen corresponding to the plain text.
6. The user can select and update the organization and role information for the end user who can view the encrypted information through the CryptoPaper scanner.
7. After all previous process is done, the user can now click the upload button. The key he/she used to encrypt the data will be uploaded to our CryptoPaper Database.
8. User can print the document out using Microsoft default function, seal it or give it to other people for further usage.

## **3.2 QR code choices**

### **3.2.1 QR-code properties**

#### **3.2.1.1 High data capacity**

The data encoded can include numbers, alphanumeric characters symbols, text symbols such as kanji (Japanese language symbols) as well as control codes, because these codes are stored both horizontally and vertically. In fact QR codes can hold text messages, website address, contact information, phone numbers and more. Table 3.1 lists the data storage capacity of QR code.

<b>Data Type</b>	<b>Maximum data amount</b>
<b>Numeric</b>	<b>7089 characters</b>
<b>Alphanumeric</b>	<b>4296 characters</b>
<b>Bytes</b>	<b>2953 characters</b>
<b>Kanji</b>	<b>1817 characters</b>

**Table 3.1.** QR code data storage capacity

### 3.2.1.2 High data density

Table 3.2 lists the data density of QR code. Since QR Code carries information both horizontally and vertically, QR Code is capable of encoding the same amount of data in approximately one-tenth the space of a traditional barcode. Paper document in general format with font calibri, font size 11 and 1.5-line spacing would have a data density of 200 bits/inch\*inch. From the table we can found that, version 2 QR code already contains the same data density as well as the normal paper format.

	<b>Capacity(bits)</b>	<b>Area(inch<sup>2</sup>)</b>	<b>Data density(bits/inch<sup>2</sup>)</b>
Paper document	19400	97.11	200
QR code (version=1)	72	0.4225	170
QR code (version=2)	128	0.5625	228
QR code (version>1)	...	...	>200

**Table 3.2.** QR code data density

### 3.2.1.3 Data restoration

Occasionally QR codes and bar codes become damaged or they may get dirty. Barcode reader will not be able to scan a damaged or dirty code. QR code can be scanned up to 30% of code words in a QR code can be restored depending upon amount of damage. Finally QR code superior in recovering lost or damage data. Ta-

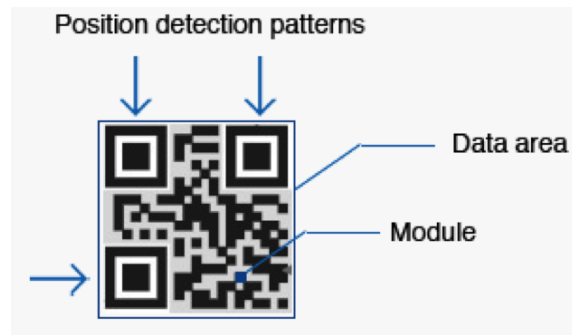
ble 3.3 lists the data restoration rate for different QR code error correction levels.

ECC Level	Data restoration rate for total
Level L	Approximate 7%
Level M	Approximate 15%
Level Q	Approximate 25%
Level H	Approximate 28%

**Table 3.3.** QR code error correction level

#### 3.2.1.4 Omni-directional and high-speed reading

QR code can be scanned from any position. This is due to the three position detection patterns located in three corner of the code. The reader will locate these three detection patterns and know how to correctly read the code. This feature speeds up the time needed to scan objects. Figure 3.2 lists the position detection patterns of QR code.

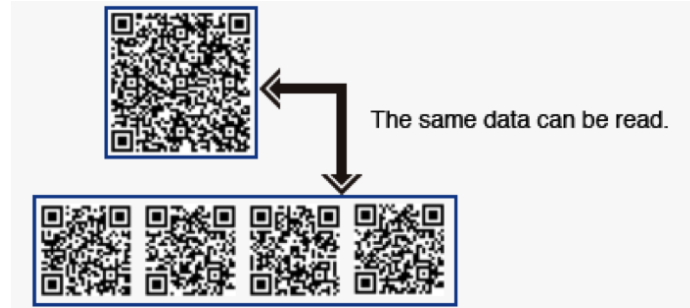


**Figure 3.2.** Position detection patterns to enable the omni-directional scan

#### 3.2.1.5 Structure Appended Feature

As the Figure 3.3 shown, QR code supports append feature. A larger QR code can be divided into as many as 16 smaller squares. This feature allow larger QR code to be stretched out on an object. Thus, larger code printed onto a narrow area. QR code located on any object.





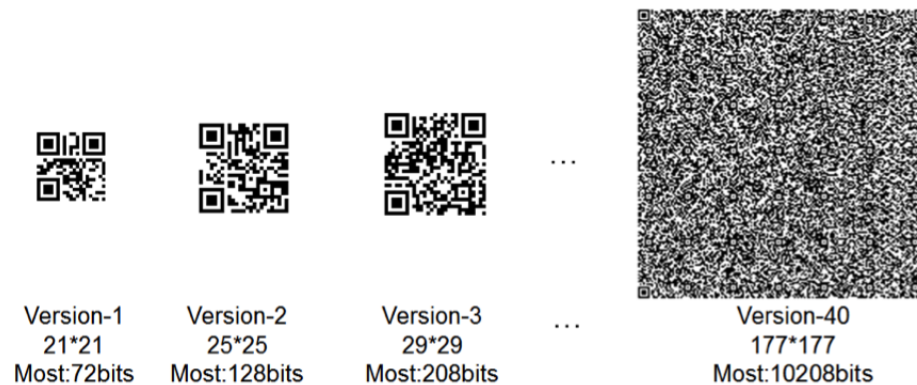
**Figure 3.3.** Structure Append Feature

### 3.2.1.6 Multiple version and size choices

One of the most important factors for us to consider when choosing the version of QR code is data density. Since the space on paper documents is limited, we should choose a version of QR code, which can encode the largest number of data bits for the same area. The formula for total information in fixed area is:

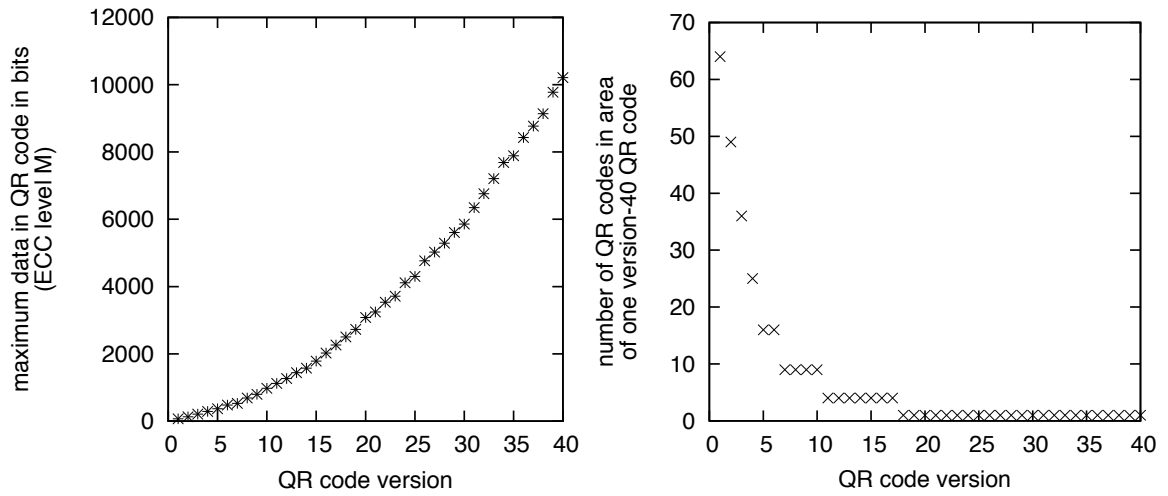
$$Total\_data\_amount = (Number\_of\_bits\_in\_one\_code) \times (Number\_of\_codes)$$

First we consider the upper bound of the number of bits in QR codes of different versions with ECC level of high, which can be shown below in Figure 3.4:

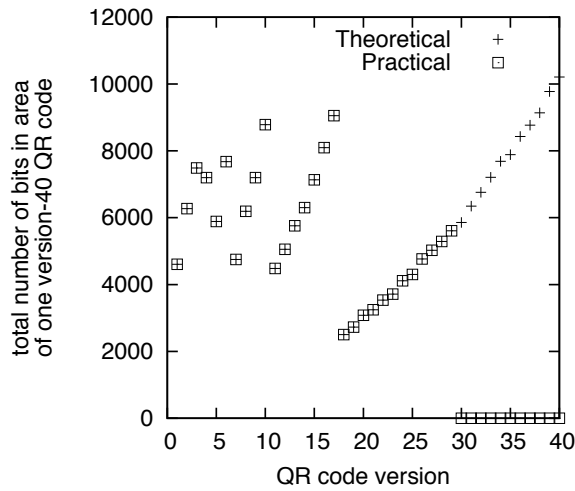


**Figure 3.4.** QR-code version properties.

As the version of QR code grows, the capacity of QR code becomes bigger. Example of QR codes are shown in Figure 3.5. The least number of bits are 72 for version-1



**Figure 3.5.** Data bits in QR codes of different versions. **Figure 3.6.** Number of QR codes that fit in space of version-40 QR code.



**Figure 3.7.** Data bits in area of version-40 QR code (theoretical and practical when read with scanner).

while the most number of bits are 10208 for version-40. Next we need to decide on how many QR codes can be inserted into the fixed area for different versions. We choose the fixed area the same as the area of QR code version 40. This can be seen on the left chart of the following Figure 3.6. By calculating the total number of bits for this fixed area using the formula above, we can get the relation between total number of bits and QR code version in the right part of figure above. There is a difference between the theoretical calculation and practical detection because our camera won't be able to recognize the QR code when the version is more than 30. Thus in this situation the total number of bits should be zero. As we can see from Figure 3.7, we see that we get the largest capacity when we choose QR code of version 3, 6, 10 or 17. These versions of QR codes can contain the largest number of bits for fixed area. In this way, we choose QR code version 17 in general for CryptoPaper system.

### **3.2.2 QR-code experiments**

#### **3.2.2.1 Printer Head Density and Module size**

The density of the dots in the printer head, number of dots per square inch, will determine the module size. For example, if the head density is 300dpi and each module is made up to 5 dots, the module size is 0.42mm. Increasing the number of dots improves the printing quality, eliminates printing width or paper feed speed fluctuations, distortion of axis, blurring, etc, and enables more stable operations. It is recommended for stable operations that each module is made up to 4 or more dots. Table 3.4 shows the relation between printer and module size.

#### **3.2.2.2 Scanner factor**

Each scanner has its own readable module size limit. The scanner resolution represents the limit. If a data code symbol is printed with a 300 dpi, 4-dot printer, the module size is 0.34mm. A scanner resolution of less than 0.34mm is required to

Printer and Module Size				
Printer	Head Density	4 Dot Module	5 Dot Module	6 Dot Module
Laser	600dpi (24dot/mm)	0.17mm	0.21mm	0.25mm
	360dpi (14dot/mm)	0.28mm	0.35mm	0.42mm
Thermal	300dpi (12dot/mm)	0.33mm	0.42mm	0.5mm
	200dpi (8dot/mm)	0.5mm	0.63mm	0.75mm



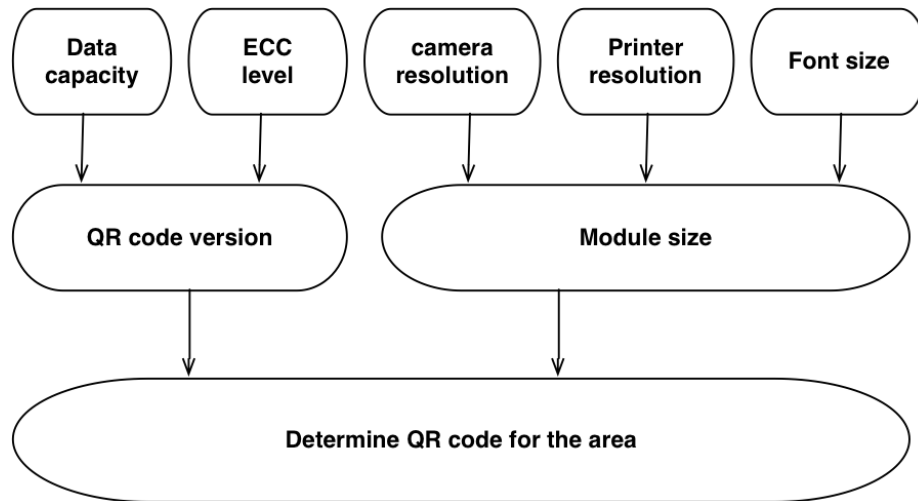
**Table 3.4.** Printer dpi and module size

read the symbol. For modern smartphones, the scanning resolution is not a problem, as the camera has a resolution exceeding most high end QR scanners.

### 3.2.2.3 QR code decision factors

The Figure 3.8 below illustrate the decisions that have to be made and the factors to consider when determining the size and version of QR code will be for a fixed area. Data capacity controls the QR code version, however, as the requirement of error correction level gets higher, the same QR code version cannot contain the same amount of data. In this way, people have to increase the QR code version. At the same time, QR code system size will become larger as the version increases. On the other hand, printer resolution and scanner camera resolution are also important factors determining the symbol size. QR code symbol size depends on the size of the individual black and white dots to be printed. The larger the dots are, the easier the code is to read and more stable the code is, but a larger printing area is also required. In this way, there is a trade-off between QR code version and ECC correction level with symbol size, given the fixed data amount and area.

My experiments use a high-resolution printer, HP Laserjet P2550, with a 8mega pixel camera on iPhone 5 to read the QR code. Also, I choose the binary encoding



**Figure 3.8.** QR code generation factors

mode for error correction level L and level H, calculate QR code size for different versions and the maximum scan distance to differentiate the code.

### 3.2.3 Experiments result analysis

Experiments result of error correction level L shown in the following Table 3.5 and Table 3.6. And experiments result of error correction level H shown in the following Table 3.7 and Table 3.8.

According to the experimental results I grow the plot to show the relations between different attributes of QR codes. Figure 3.9 shows that QR code size increase linearly with the QR code version. A Version 1 QR code will have 21 rows and 21 columns of dots and the version number will then increase by 1 for every 4 extra rows and columns - a Version 2 QR code will have 25 rows and columns, a Version 3 will have 29 rows and columns, right through to a Version 25 that will have 117 rows and columns. And code size will increase by 4 times of module size. Figure 3.10 shows that maximum scan distance is proportional to the QR code size. For the same error correction level code with same version number, the scan distance is of 1.0mm module size is twice the scan distance of 0.5mm module size. That means as long as we enlarge the QR

Version	Max Characters	ECC level	Module size (0.5mm)	Size	Scan distance
1	17	L	21*21	10.5mm	160mm
2	32	L	25*25	12.5mm	158mm
3	53	L	29*29	14.5mm	160mm
4	78	L	33*33	16.5mm	161mm
5	106	L	37*37	19.0mm	170mm
6	134	L	41*41	21.0mm	165mm
7	154	L	45*45	23.0mm	165mm
8	192	L	49*49	25.5mm	165mm
9	230	L	53*53	27.5mm	165mm
10	271	L	57*57	29.5mm	165mm
11	321	L	61*61	31.5mm	170mm
12	367	L	65*65	33.5mm	170mm
13	425	L	69*69	36.0mm	175mm
14	458	L	73*73	38.0mm	180mm
15	520	L	77*77	40.0mm	180mm
16	586	L	81*81	42.5mm	180mm
17	644	L	85*85	44.5mm	180mm
18	718	L	89*89	46.5mm	180mm
19	792	L	93*93	48.5mm	180mm
20	858	L	97*97	50.5mm	185mm
21	929	L	101*101	53.0mm	190mm
22	1003	L	105*105	55.0mm	190mm
23	1091	L	109*109	57.0mm	190mm
24	1171	L	113*113	59.0mm	190mm
25	1273	L	117*117	61.5mm	190mm
26	1367	L	121*121	63.5mm	195mm
27	1465	L	125*125	65.5mm	195mm
28	1528	L	129*129	67.5mm	200mm
29	1628	L	133*133	70.0mm	200mm
30	1732	L	137*137	72.0mm	200mm
31	1840	L	141*141	74.0mm	200mm
32	1952	L	145*145	76.0mm	205mm
33	2068	L	149*149	78.5mm	205mm
34	2188	L	153*153	80.5mm	205mm
35	2303	L	157*157	82.5mm	210mm
36	2431	L	161*161	84.5mm	210mm
37	2563	L	165*165	87.0mm	210mm
38	2699	L	169*169	89.0mm	210mm
39	2809	L	173*173	91.5mm	210mm
40	2953	L	177*177	93.5mm	210mm

**Table 3.5.** Test distance for 0.5mm module size with level L

Version	Max Characters	ECC level	Module size (1mm)	Size	Scan distance
1	17	L	21*21	21.5mm	325mm
2	32	L	25*25	25.5mm	325mm
3	53	L	29*29	29.5mm	330mm
4	78	L	33*33	33.5mm	330mm
5	106	L	37*37	38.5mm	330mm
6	134	L	41*41	42.5mm	330mm
7	154	L	45*45	46.5mm	335mm
8	192	L	49*49	51.5mm	335mm
9	230	L	53*53	55.5mm	335mm
10	271	L	57*57	59.5mm	335mm
11	321	L	61*61	64.5mm	340mm
12	367	L	65*65	68.0mm	340mm
13	425	L	69*69	72.0mm	345mm
14	458	L	73*73	76.5mm	350mm
15	520	L	77*77	80.5mm	355mm
16	586	L	81*81	85.0mm	355mm
17	644	L	85*85	89.0mm	360mm
18	718	L	89*89	93.5mm	360mm
19	792	L	93*93	97.5mm	360mm
20	858	L	97*97	102.0mm	370mm
21	929	L	101*101	106.0mm	380mm
22	1003	L	105*105	110.0mm	380mm
23	1091	L	109*109	115.0mm	380mm
24	1171	L	113*113	118.5mm	380mm
25	1273	L	117*117	123.0mm	385mm
26	1367	L	121*121	127.0mm	390mm
27	1465	L	125*125	132.0mm	390mm
28	1528	L	129*129	136.0mm	400mm
29	1628	L	133*133	140.5mm	400mm
30	1732	L	137*137	144.5mm	400mm
31	1840	L	141*141	149.0mm	400mm
32	1952	L	145*145	153.0mm	410mm
33	2068	L	149*149	157.0mm	410mm
34	2188	L	153*153	161.5mm	410mm
35	2303	L	157*157	165.5mm	415mm
36	2431	L	161*161	169.5mm	420mm
37	2563	L	165*165	173.5mm	420mm
38	2699	L	169*169	177.5mm	420mm
39	2809	L	173*173	181.5mm	420mm
40	2953	L	177*177	185.5mm	420mm

**Table 3.6.** Test distance for 1.0mm module size with level L

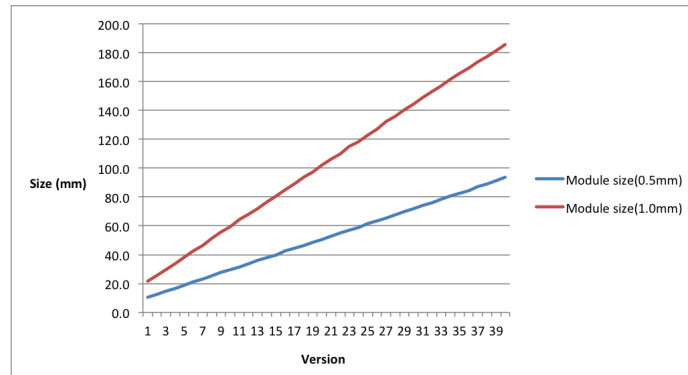
Version	Max Characters	ECC level	Module size (0.5mm)	Size	Scan distance
1	7	H	21*21	10.5mm	160mm
2	14	H	25*25	12.5mm	158mm
3	24	H	29*29	14.5mm	160mm
4	34	H	33*33	16.5mm	161mm
5	44	H	37*37	19.0mm	170mm
6	58	H	41*41	21.0mm	165mm
7	64	H	45*45	23.0mm	165mm
8	84	H	49*49	25.5mm	165mm
9	98	H	53*53	27.5mm	165mm
10	119	H	57*57	29.5mm	165mm
11	137	H	61*61	31.5mm	170mm
12	155	H	65*65	33.5mm	170mm
13	177	H	69*69	36.0mm	175mm
14	194	H	73*73	38.0mm	180mm
15	220	H	77*77	40.0mm	180mm
16	250	H	81*81	42.5mm	180mm
17	280	H	85*85	44.5mm	180mm
18	310	H	89*89	46.5mm	180mm
19	338	H	93*93	48.5mm	180mm
20	382	H	97*97	50.5mm	185mm
21	403	H	101*101	53.0mm	190mm
22	439	H	105*105	55.0mm	190mm
23	461	H	109*109	57.0mm	190mm
24	511	H	113*113	59.0mm	190mm
25	535	H	117*117	61.5mm	190mm
26	593	H	121*121	63.5mm	195mm
27	625	H	125*125	65.5mm	195mm
28	658	H	129*129	67.5mm	200mm
29	698	H	133*133	70.0mm	200mm
30	742	H	137*137	72.0mm	200mm
31	790	H	141*141	74.0mm	200mm
32	842	H	145*145	76.0mm	205mm
33	898	H	149*149	78.5mm	205mm
34	958	H	153*153	80.5mm	205mm
35	983	H	157*157	82.5mm	210mm
36	1051	H	161*161	84.5mm	210mm
37	1093	H	165*165	87.0mm	210mm
38	1139	H	169*169	89.0mm	210mm
39	1219	H	173*173	91.5mm	210mm
40	1273	H	177*177	93.5mm	210mm

**Table 3.7.** Test distance for 0.5mm module size with level H

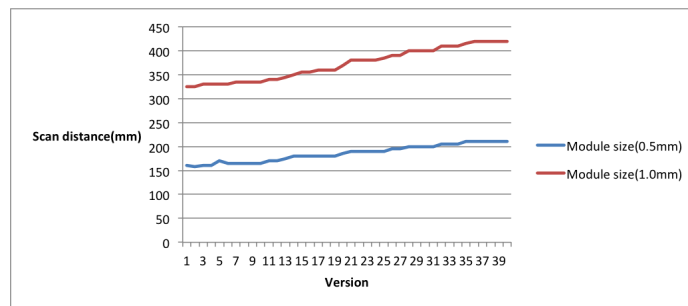
Version	Max Characters	ECC level	Module size (0.5mm)	Size	Scan distance
1	7	H	21*21	10.5mm	160mm
2	14	H	25*25	12.5mm	158mm
3	24	H	29*29	14.5mm	160mm
4	34	H	33*33	16.5mm	161mm
5	44	H	37*37	19.0mm	170mm
6	58	H	41*41	21.0mm	165mm
7	64	H	45*45	23.0mm	165mm
8	84	H	49*49	25.5mm	165mm
9	98	H	53*53	27.5mm	165mm
10	119	H	57*57	29.5mm	165mm
11	137	H	61*61	31.5mm	170mm
12	155	H	65*65	33.5mm	170mm
13	177	H	69*69	36.0mm	175mm
14	194	H	73*73	38.0mm	180mm
15	220	H	77*77	40.0mm	180mm
16	250	H	81*81	42.5mm	180mm
17	280	H	85*85	44.5mm	180mm
18	310	H	89*89	46.5mm	180mm
19	338	H	93*93	48.5mm	180mm
20	382	H	97*97	50.5mm	185mm
21	403	H	101*101	53.0mm	190mm
22	439	H	105*105	55.0mm	190mm
23	461	H	109*109	57.0mm	190mm
24	511	H	113*113	59.0mm	190mm
25	535	H	117*117	61.5mm	190mm
26	593	H	121*121	63.5mm	195mm
27	625	H	125*125	65.5mm	195mm
28	658	H	129*129	67.5mm	200mm
29	698	H	133*133	70.0mm	200mm
30	742	H	137*137	72.0mm	200mm
31	790	H	141*141	74.0mm	200mm
32	842	H	145*145	76.0mm	205mm
33	898	H	149*149	78.5mm	205mm
34	958	H	153*153	80.5mm	205mm
35	983	H	157*157	82.5mm	210mm
36	1051	H	161*161	84.5mm	210mm
37	1093	H	165*165	87.0mm	210mm
38	1139	H	169*169	89.0mm	210mm
39	1219	H	173*173	91.5mm	210mm
40	1273	H	177*177	93.5mm	210mm

**Table 3.8.** Test distance for 1.0mm module size with level H

code with the same dots configuration, the maximum scan distance will also enlarge by the same number. Following that, I draw the plot between the relation of scan



**Figure 3.9.** Relation between version and size for different modules

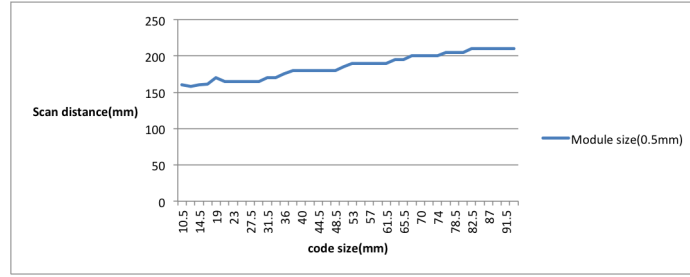


**Figure 3.10.** Relation between version and scan distance for different modules

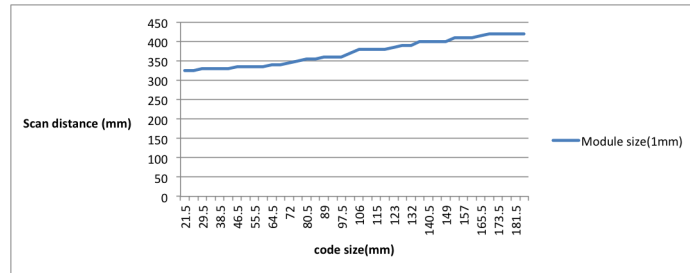
distance and code size, and testify the linear relations. As we can see in the following Figure 3.11 and Figure 3.12.

According to the above experiments, I get the conclusion that QR code size has a linear relation with its module number(version), as the version increase, code size will increase by a fixed amount. And also, scan distance depends on code size, for my configuration scan distance is nearly 15 times of QR code size. This factor can be influenced by many exterior conditions, a better camera resolution could increase this value and at the same time, a poor lighting could decrease this value.





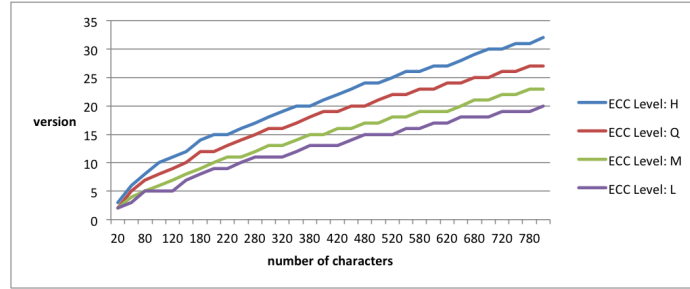
**Figure 3.11.** Relation between scan distance and size for module 1



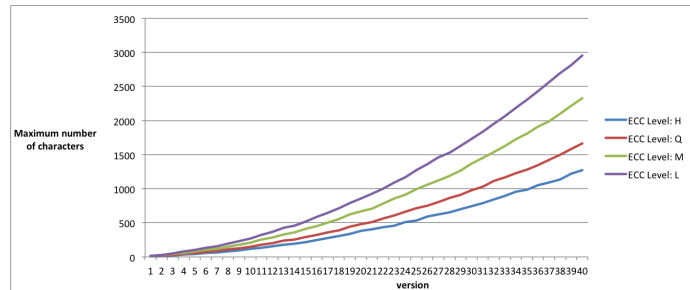
**Figure 3.12.** Relation between scan distance and size for module 2

$$Maximum\_scan\_distance = (Exterior\_factor) \times (Number\_of\_modules) \times (Minimum\_module\_size)$$

Another interesting experiments is the relation between QR code version and maximum data amounts for each error correction levels. I draw the plot Figure 3.13 and Figure 3.14, and find that level L contains the most data amount while H contains the least. Level M is commonly used as the default value for QR code generation for its good performance. Level M contains almost 80% data amount as the same version of Level L, while Level Q contains almost 56% of data, and Level H contains almost 43%. From the other view, give a fixed data amount code version with Level H is almost twice the code version of Level L.



**Figure 3.13.** Relation between scan distance and size for module 1



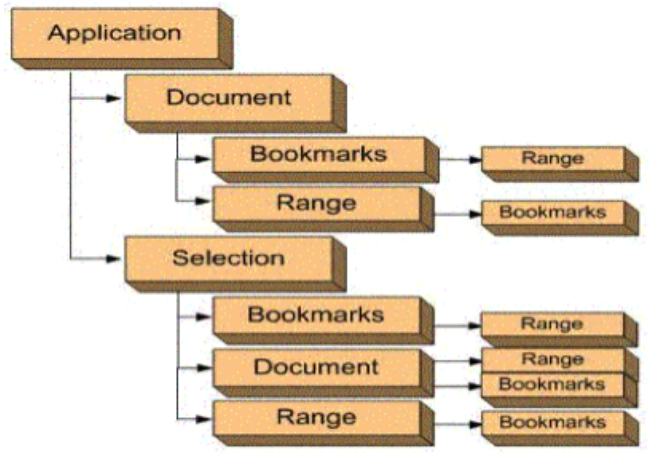
**Figure 3.14.** Relation between scan distance and size for module 2

### 3.2.4 Implementation in MS Word add-in

#### 3.2.4.1 Word Object Module in a developer's perspective

Microsoft Word is probably one of the most commonly used software products in the world today; almost every office and laboratory is equipped with it. Microsoft Visual Studio Tools provides developers a rich and powerful programming environment to enable them to interact with object model by using .NET framework. Figure 3.15 describes the Word Object Model and how they interact with each other.

**The Application Object:** The Application object represents the Word application, and is the parent of all of the other objects. Its members usually apply to Word as a whole. Developers can use its properties and methods to control the Word environment.



**Figure 3.15.** The Application object contains the Document, Selection, Bookmark, and Range objects.

The Document Object: The Document object is central to programming Word. When the user open an existing document or create a new document, he or she creates a new Document object, which is added to the Word Documents collection. The document that has the focus is called the active document and is represented by the application object's ActiveDocument property.

The Selection Object: The Selection object represents the area that is currently selected. When user performs an operation in the Word user interface, such as bolding text, he or she selects the text and then apply the formatting. The Selection object is always present in a document; if nothing is selected, the Selection object represents the insertion point. The Selection object can also be multiple noncontiguous blocks of text.

The Range Object: The Range object represents a contiguous area in a document, and is defined by a starting character position and an ending character position. It is not limited to a single Range object; developers can define multiple Range objects in the same document.

The Bookmark Object: The Bookmark object is similar to the Range object in that it represents a contiguous area in a document, with both a starting position and

an ending position. Developers use bookmarks to mark a location in a document, or as a container for text in a document. A Bookmark object can consist of the insertion point alone or be as large as the entire document. A user can also define multiple bookmarks in a document.

Our CryptoPaper system uses application-level module to control the whole Word Environment. We use this class to perform tasks such as accessing the object model of the Microsoft Office host application, customizing the user interface (UI) of the application, and connect to the Amazon RDS cloud system. And also, the selection-level module realizes the function of encrypting selected plaintext and changing it into QR-code. Also we use application-level object to customize the ribbon, display a custom task pane for the UI design.

#### **3.2.4.2 Introduction to the QR-code library in c#**

In our CryptoPaper project, QR-code will be printed on traditional paper, which then can be captured by the scanner to recover the original information. The recovered information is then displayed if the user is authorized to view the information. We chose to use open source QR-code library called Messagingtoolkit to encode our cipher text into QR-code. This QR-code library is a .NET component that can be used to encode and decode QR-codes. It provides several properties that are used to define a QR-code. It also provides functions to encode content into a QR-code image, which can be saved in JPEG, GIF, PNG, or BITMAP formats and decode a QR-code image. QR code encoders provide a number of improved features over conventional barcodes which include high capacity encoding, improved efficiency of information storage, capabilities to store Kana and Kanji characters, damage resistance, Omni-directional encoding and multiple data areas. Developers can choose properties such as encoding modes, error correction level and code version for different image formats.

In CryptoPaper system, we choose alphanumeric modes, high-level error correction and version 17 in general.

### 3.2.4.3 Select the best choices among trade-offs

Version \ Characters	ECC Level: M	ECC Level: L	ECC Level: Q	ECC Level: H
4	15	30	7	0
7	60	90	30	5
9	110	140	60	45
11	150	200	110	75
13	210	270	150	105
15	280	350	190	145
17	350	440	230	180
19	430	540	280	220
21	510	650	340	260
23	620	780	400	310
25	710	920	470	365
27	810	1070	560	420
29	920	1200	650	480
31	1050	1350	750	550
33	1190	1510	840	630
35	1320	1680	920	700
37	1470	1860	1020	795
40	1720	2150	1210	925

**Table 3.9.** QR code version for fixed data amount of different ECC levels

Except for the encrypted information, code id and other meta information totally consumes 25 character. Also I take into consideration characters out of ASCII will take extra spaces. I test version 4, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 40 for each error correction levels and get the result as follows table 3.9. Because in our case, level 4 is the minimum requirement for containing the basic information including meta information. Then I choose every two steps from level 7 to level 37 for each version, and finally test the version 40 for the maximum data amount.

I designed the following mechanism of choosing appropriate codes in different scenarios. Initially, according to the font size, system selects corresponding module size, and set up the default error correction level as M. If the user choose a higher error correction level, given selected data amount, our system could choose a higher version in order to contains the data. On the contrary, if the user choose a lower level

error correction, system would automatically choose a lower version. If data amount exceeds the upper bound, a system prompt would jump out to remind the user, QR code cannot contains so much data.

### 3.3 AES encryption

AES is a cryptographic algorithm that often used to protect electronic data. Specifically, AES is an iterative, symmetric-key block cipher that can use keys of 128, 192, and 256 bits, and encrypts and decrypts data in blocks of 128 bits (16 bytes). Figure 5 describes the process of encryption of AES. Unlike public-key ciphers, which use a pair of keys, symmetric-key ciphers use the same key to encrypt and decrypt data. Encrypted data returned by block ciphers have the same number of bits that the input data had.

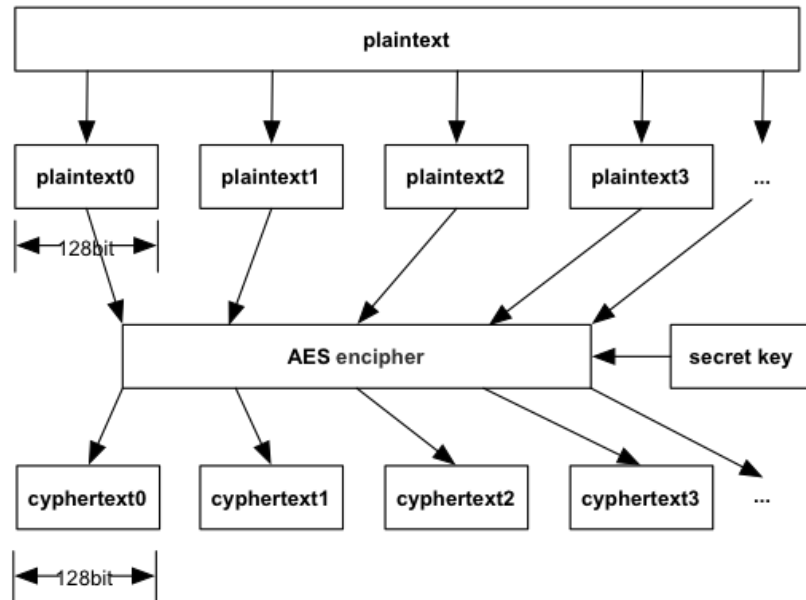


Figure 3.16. AES encryption in ECB mode.

We use the Electronic Codebook (ECB) mode encryption to encrypt each block individually. Any blocks of plaintext that are identical and in the same message, or that are in a different message encrypted with the same key, will be transformed

into identical cipher text blocks. Even though this mode opens the door for multiple security exploits, we still choose this since this is a faster encryption mode over the other modes. In order to further secure it we use additional layers of security like encoding the cipher text into QR codes and applying a digital signature which makes it difficult for the attacker to obtain the data or modify it. The mode will enhance user experience by decreasing overall latency time in querying the database and the reaction speed of the scanner. Figure 3.16 describes the AES encryption in ECB mode.

### **3.4 Microsoft MySQL database connectivity**

MySQL connector/Net provides us an easy environment to develop .NET applications that require secure, high-performance data connectivity with MySQL. It implements the required ADO.NET interfaces and integrates into ADO.NET-aware tools. We use it to build the connectivity between MySQL database on Amazon cloud and our word plugin.

We use many projects of MySQL connector/NET library to work with data. In the following, below listed are several primary objects that we use in CryptoPaper: The SqlConnection Object: The SqlConnection object helps identify the database server, the database name, user name, password, and other parameters that are required for connecting to the database.

The SqlCommand Object: The SqlCommand object is used for interacting with MySQL database. We use a command object to send SQL statements to the database. A command object uses a connection object to figure out which database to communicate with.

The SqlDataReader Object: The data reader object allows us to obtain the results of a SELECT statement from a command object. For performance reasons, the data

returned from a data reader is a fast forward-only stream of data. This ensures the speed that we pull the data from the stream in a sequential manner.

**The DataSet Object:** DataSet objects are in-memory representations of data. They contain multiple data table objects, which contain columns and rows, just like normal database tables. Also, we use it define relations between tables to create parent-child relationships. The DataSet is specifically designed to help manage data in memory and to support disconnected operations on data, when such a scenario make sense.

**The SqlDataAdapter Object:** When the data we work with is primarily read-only and we need to make changes to the underlying data source, we use Data Adapter Object. Some situations also call for caching data in memory to minimize the number of database calls for data that does not change. The data adapter makes it easy to accomplish these things by helping to manage data in a disconnected mode. The data adapter fills a DataSet object when reading the data and writes in a single batch when persisting changes back to the database. A data adapter contains a reference to the connection object and opens and closes the connection automatically when reading from or writing to the database. Additionally, the data adapter contains command object references for SELECT, INSERT, UPDATE, and DELETE operations on the data.

### **3.5 Password hash**

The most important aspect of a user account system is how user passwords are protected. User account databases are hacked frequently, so when I create this database I must do something to protect user's passwords once our system is breached. The best way to protect passwords is to employ salted password hashing.

Hash algorithms are one-way functions. They turn any amount of data into a fixed-length “fingerprint” that cannot be reversed. They also have the property that



if the input changes by even a tiny bit, the resulting hash is completely different (see the example above). This is great for protecting passwords, because we want to store passwords in a form that protects them even if the password file itself is compromised, but at the same time, we need to be able to verify that a user's password is correct. The general workflow for account registration and authentication in a hash-based account system is as follows:

1. The user creates an account.
2. Their password is hashed and stored in the database. At no point is the plain-text (unencrypted) password ever written to the hard drive.
3. When the user attempts to login, the hash of the password they entered is checked against the hash of their real password (retrieved from the database).
4. If the hashes match, the user is granted access. If not, the user is told they entered invalid login credentials.
5. Steps 3 and 4 repeat every time someone tries to login to their account.

Also, never remind the user if it was the username or password they got wrong. Always display a generic message like "Invalid username or password". This prevents attackers from enumerating valid usernames without knowing their passwords. If two users have the same password, they'll have the same password hashes if no salt exists. Attackers can use lookup tables or dictionary attack to crack the database. So I append or prepend a random string, called salt, to randomize the hashes of password before hashing. This makes the same password hash into a completely different string every time. To check if a password is correct, users need the salt, so it is usually stored in the user account database along with the hash, or as part of the hash string itself. The salt does not need to be secret. Just by randomizing the hashes, lookup tables, reverse lookup tables, and rainbow tables become ineffective. An attacker won't know

in advance what the salt will be, so they can't pre-compute a lookup table or rainbow table. If each user's password is hashed with a different salt, the reverse lookup table attack won't work either. So I generate salt using Cryptographically Secure Pseudo-Random Number Generator (CSPRNG). Then prepend the salt to the password and hash it with a standard cryptographic hash function such as SHA256. After that, save both the salt and the hash in the user's database record. When the user try to login, our system will retrieve the user's salt and hash from the database. Prepend the salt to the given password that user input and hash it using the same hash function. At last, comparing the hash of given password with the hash from the database. If they match, the password correct and user logged-in, otherwise a reminder like "invalid username or password" will show up.

## CHAPTER 4

### SYSTEM TEST AND EVALUATION

After designing the word add-in, I created four test cases to test different functionalities of it.

Font size test, as the Figure 4.1 shows, there are three different font sizes for the same error correction level. Font size 16 has module size 0.25mm code, when font size 20 has module size 0.5mm code and font size 32 has module size 0.75mm code. Every time I generate a code, the system will notify the code information with version, number of characters and ECC level. The result passed the test, as the Figure 4.2 shows.

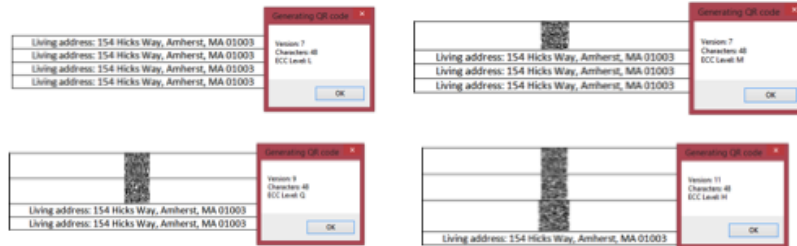


**Figure 4.1.** Test case for different font size



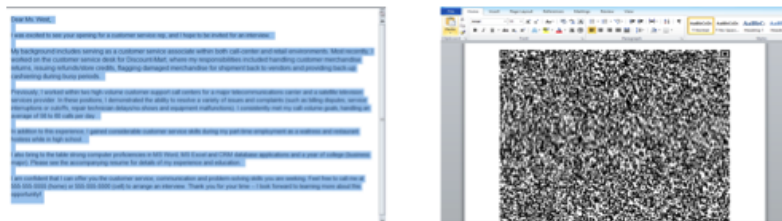
**Figure 4.2.** Font size test result

Different error correction level test, for the same data, I choose four different level correction levels. And I find that level L and level M requires version 7 for 48 characters, while level Q requires version 9 and level H requires version 11. The process was recorded in the Figure 4.3, and it passed the test.



**Figure 4.3.** Test case for long paragraphs

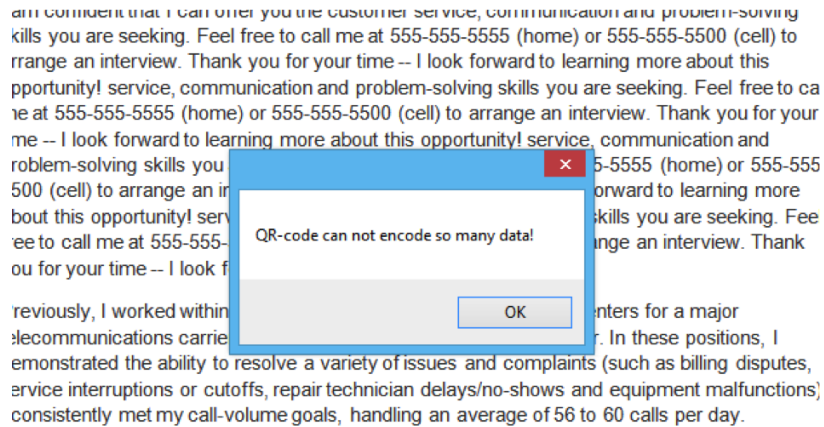
A long paragraph test, I use the default error correction level to encode the whole paragraph of a cover letter, as much as 1000 characters. It successfully encode the data into a version 40 QR code, as the Figure 4.4 shows.



**Figure 4.4.** Test case for long paragraphs

Data overflow test, if the data amount exceeds the maximum value of current chosen error correction level, our system promo will jump out to notify the user as the Figure 4.5 shows.

After test the functionality of word add-in, I also test the whole system functionality with different access right. By using word add-in, I generate the a paper document as the Figure 4.6 shown below, replace home phone and social security number with QR code, set up different access control for each code. After that, I



**Figure 4.5.** Test case for data overflow

printed it out, using desktop java application to scan the code. If I log-in using a professor's identity, since home phone is accessible, I can see the phone number through the scanner screen. However, since social security number is only accessible by the HR in UMASS, I can't recover that part of data through the scanner. This result shown in Figure 4.7 passed the test.

Inception of the idea of developing a CryptoPaper system started due to negligence of people about physical document security. Through the ideas of Japanese scientists, we came out of this solution. It provides a hybrid solution to combined digital security properties with common physical documents. Decisions regarding various tools and designs used in developing the system were based on faster implementation and security principles. Potential risks like man-in-the middle attacks, system crashes were reduced to the minimum. Our system realize confidentiality and authenticity security properties using a simplified meta-information structure. Based on this process and the properties that it achieves, CryptoPaper can be used in a range of application scenarios such as healthcare documents, bank transaction documents, or human resource documents. For example, HR people in each company take control sensitive information of each employee such as social security number, salary etc. Once introduced our system, employees can encode their privacies before


**Employee Information Form**

**Personal Information**


Full Name: **Tony Stark**  
Last First M.I.

Address: **10880 Malibu Point 001**  
Street Address Apartment/Unit #

**Malibu Point California 90265**  
City State ZIP Code

Home Phone:  Alternate Phone: **No**

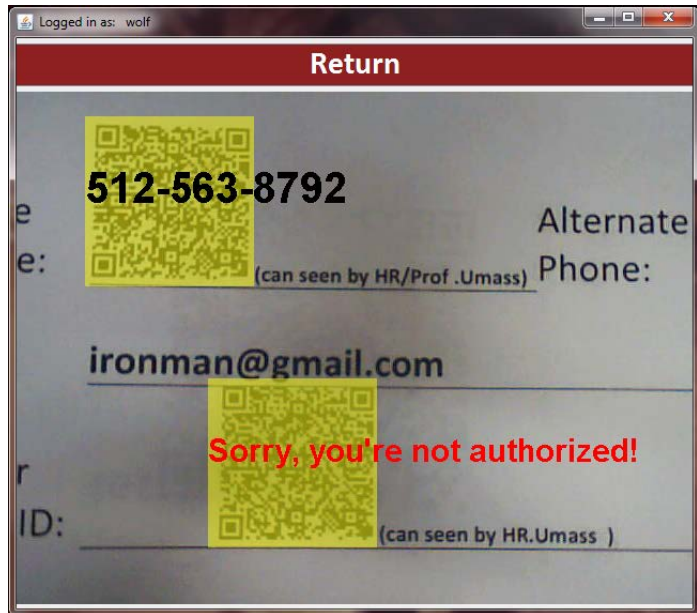
Email: **ironman@gmail.com**

SSN or Gov't ID: 

Birth Date: **06/24/1972** Marital Status: **Married**

Spouse's Name: **Pepper Potts**

**Figure 4.6.** Generated employee information form using word add-in



**Figure 4.7.** Test result for access control of CryptoPaper system

submit document to HR people, and set up the access right of that document to HR only. Thus they don't have to worry about their personal information leakage due to careless of paper document storage.

## CHAPTER 5

### SUMMARY

Information, especially confidential information, is highly valued by the government, companies and individuals. Therefore, sensitive information on paper documents need to be secured. In this project, we design and implement a prototype evaluation of system called CryptoPaper. It replaces sensitive information on paper documents with QR-codes and let only authorized user can decode it with our scanner. It realizes confidentiality and authenticity security properties on physical paper, uploads key and access right information into cloud database and leave real data on paper. We present a prototype system implementation that generates codes using a Microsoft Word plugin, stores cryptographic keys in a scalable cloud database, and uses an embedded system scanner to read and display decoded information. The presented system can be applied to a range of real-world applications that involve sensitive information.

I focused on CryptoPaper generation part, learnt about different attributes of 2D codes, especially QR code. Then I did a series of experiments to explore the relations between different attributes of QR code. And utilize the experiment results to design and implement MS word add-in, let our system generate the best QR code choice for different scenarios.



## BIBLIOGRAPHY

- [1] Taizo Anan, Kensuke Kuraki, Jun Takahashi *Paper Encryption Technology* 2010: FUJITSU Sci. Tech. J., Vol. 46, No. 1, pp. 87-94
- [2] California State University, 2D Barcodes Information and Guidelines
- [3] Amandeep Kaur, Harwinder Sohal, *QR Code Library on the Base of Software Reuse Approach*, International Journal of Science and Engineering Applications (IJSEA) Volume 2 Issue 3, 2013, ISSN - 2319-7560 (online)
- [4] Mr. Nachiket A. Rathod, Dr. Siddharth A. Ladhake Detecting and Decoding Algorithm for 2D Barcode 2012: International Journal of Emerging Technology and Advanced Engineering
- [5] *INTERNATIONAL STANDARD ISO/IEC 18004* Information technology - Automatic identification and data capture techniques - QR Code 2005 bar code symbology specification
- [6] NPO Japan Network Security Association: FY2008 Investigation Report on Information Security Incidents (Ver.1.3). (in Japanese)
- [7] MSDN magazine, Eric Faller, The 2007 Office System With Your Own Ribbon Tabs And Controls, <https://msdn.microsoft.com/en-us/magazine/cc163469.aspx>
- [8] MSDN magazine, James McCarey, Keep Your Data Secure with the New Advanced Encryption Standard, <https://msdn.microsoft.com/en-us/magazine/cc164055.aspx>
- [9] CodeProject, First VSTO Application, <http://www.codeproject.com/Articles/192724/Justin-s-VSTO-Knowledge-Base-First-VSTO-Applica>
- [10] Omar Al Zabir, "Implement a Microsoft Word-like Object Model for Your .NET Framework Application" 2005, <http://msdn.microsoft.com/en-us/library/ms973253.aspx>
- [11] Taizo Anan et al.: Watermark Technology Realizing Security of Printed material (in Japanese) 2007, FUJITSU, Vol. 58, No. 3, pp. 183-187
- [12] Amazon RDS service: user guide (API version 2014-10-31), 2014

- [13] Dynamo: Amazon’s Highly Available Key-value Store, 2007 ACM 978-1-59593-591-5/07/0010
- [14] *Fast Detection and Recognition of QR codes in High-Resolution Images* Szentandrsi, Istvn, Adam Herout, and Markta Dubsk, Proceedings of the 28th Spring Conference on Computer Graphics. ACM, 2012.
- [15] Adobe Geoff.Baum, Bridging the paper-to-digital divide with 2D barcode technology, 2005
- [16] CrackStation, Salted Password Hashing - Doing it Right, 2014, <https://crackstation.net/hashing-security>
- [17] twit88, “messaging toolkit open source QRCode Library” 2007, <http://www.codeproject.com/Articles/20574/Open-Source-QRCode-Library>.
- [18] S. Owen, “Zxing (“Zebra Crossing”),” 2013, <https://code.google.com/p/zxing/>.
- [19] Viral Patel, How To Create QR Codes In Java 2012, <http://viralpatel.net/blogs/create-qr-codes-java-servlet-qr-code-java/>